SQL JOIN and different types of JOINs

I have been going through many threads on **SO** and some other forums. So I thought I would summarize "**What is SQL JOIN?**" and "**What are different types of SQL JOINs?**".

sql | database | oracle | postgresql | rdbms

asked Jul 30 '13 at 11:48

M-D
1,078 2 10 20

1 @ClodoaldoNeto - StackExchange encourage users to do so, see this blog post . There is also a lot of posts from this kind, like this, this and others. – Mahmoud Gamal Jul 30 '13 at 12:07

@Mahmoud That post is about answering your own question. This **question** shows no research effort and its only purpose is to be answered by a tutorial answer. – Clodoaldo Neto Jul 30 '13 at 15:47

1 @ClodoaldoNeto: I think what I did is perfectly valid on SO. Thats why they have put a "Answer your own question" option. I just wanted to let others know in just a few minutes, which I took hours to organize, going through many different sites. – M-D Jul 30 '13 at 16:52

1 *which I took hours to organize* The manual is ready and **correct** and has a tutorial – Clodoaldo Neto Jul 30 '13 at 17:06

for better explanation see wiki – shreyas Nov 11 '13 at 12:52

show **1** more comment

3 Answers

What is SQL JOIN ?

SQL JOIN is a method to retrieve data from two or more database tables.

What are the different SQL JOINs ?

There are a total of five JOINs. They are :

1. **JOIN or INNER JOIN**
2. **OUTER JOIN**

   2.1 **LEFT OUTER JOIN or LEFT JOIN**
   2.2 **RIGHT OUTER JOIN or RIGHT JOIN**
   2.3 **FULL OUTER JOIN or FULL JOIN**

3. **NATURAL JOIN**
4. **CROSS JOIN**
5. SELF **JOIN**

1. JOIN or INNER JOIN :

In this kind of a JOIN, we get all records that match the condition in both the tables, and records in both the tables that do not match are not reported.

In other words, INNER JOIN is based on the single fact that : ONLY the matching entries in BOTH the

tables SHOULD be listed.

Note that a JOIN without any other JOIN keywords (like INNER, OUTER, LEFT, etc) is an INNER JOIN. In other words, INNER JOIN is a Syntactic sugar for JOIN (see : Difference between JOIN and INNER JOIN).

2. OUTER JOIN :

Outer Join retrieves

Either, the matched rows from one table and all rows in the other table Or, all rows in all tables (it doesn't matter whether or not there is a match).

There are three kinds of Outer Join :

2.1 LEFT OUTER JOIN or LEFT JOIN

This join returns all the rows from the left table in conjunction with the matching rows from the right table. If there are no columns matching in the right table, it returns NULL values.

2.2 RIGHT OUTER JOIN or RIGHT JOIN

This join returns all the rows from the right table in conjunction with the matching rows from the left table. If there are no columns matching in the left table, it returns NULL values.

2.3 FULL OUTER JOIN or FULL JOIN

This join combines left outer join and right outer join. It returns row from either table when the conditions are met and returns null value when there is no match.

In other words, OUTER JOIN is based on the fact that : ONLY the matching entries in ONE OF the tables (RIGHT or LEFT) or BOTH of the tables(FULL) SHOULD be listed.

```
Note that OUTER JOIN is a loosened form of INNER JOIN.
```

3. NATURAL JOIN :

It is based on the two conditions :

1. the JOIN is made on all the columns with the same name for equality.
2. Removes duplicate columns from the result.

This seems to be more of theoretical in nature and as a result (probably) most DBMS don't even bother supporting this.

4. CROSS JOIN :

It is the Cartesian product of the two tables involved. The result of a CROSS JOIN will not make sense in most of the situations. Moreover, we wont need this at all (or needs the least, to be precise).

5. SELF JOIN :

It is not a different form of JOIN, rather it is a JOIN (INNER, OUTER, etc) of a table to itself.

JOINs based on Operators

Depending on the operator used for a JOIN clause, there can be two types of JOINs. They are

1. Equi JOIN
2. Theta JOIN

1. Equi JOIN :

For whatever JOIN type (INNER, OUTER, etc), if we use ONLY the equality operator (=), then we say that the JOIN is an Equi JOIN.

2. Theta JOIN :

This is same as Equi JOIN but it allows all other operators like >, <, >= etc.

> Many consider both Equi JOIN and Theta JOIN similar to INNER, OUTER etc JOINs. But I strongly believe that its a mistake and makes the ideas vague. Because INNER JOIN, OUTER JOIN etc are all connected with the tables and their data where as Equi JOIN and Theta JOIN are only connected with the operators we use in the former.
>
> Again, there are many who consider NATURAL JOIN as some sort of "peculiar" Equi JOIN. In fact, it is

true, because of the first condition I mentioned for NATURAL JOIN. However, we dont have to restrict that simply to NATURAL JOINs alone. INNER JOINs, OUTER JOINs etc could be an Equi JOIN too.

answered Jul 30 '13 at 11:48

M-D
1,078 2 10 20

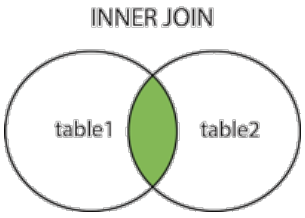
2 There are relatively new LATERAL JOIN .. SELECT * FROM r1, LATERAL fx(r1) – Pavel Stehule Jul 30 '13 at 11:52

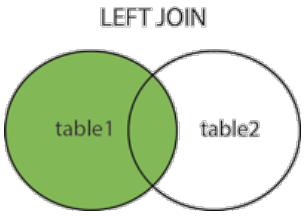
2 While this seems reasonable, I don't think answers "what is an SQL join" in any way that conveys useful information. The answer as a whole is a reference written for people who already understand joins, not for the sorts of people who are asking those questions. It also omits references, both to support its claims (as is appropriate if making an authorative answer) and to provide additional explanation via external resources. If you're trying to write an authoritative answer to link new SQL users to, it might be worth filling in the blanks a bit, especially the "what is a join" part. – Craig Ringer Jul 30 '13 at 12:55

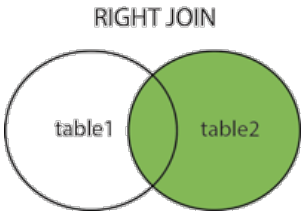
1 @CraigRinger ARE YOU SERIOUS ? – tony9099 Mar 7 at 10:39

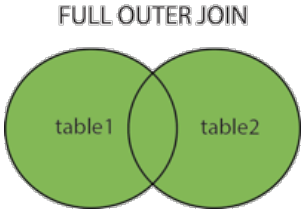
add comment

Better Illustration over theory









answered Nov 30 '13 at 9:34


Anup
409 2 6

6 A picture is worth a thousand words! – Alyas Apr 11 at 7:52

add comment

Definition:

JOINS are way to query the data that combined together from multiple tables simultaneously.

Types of JOINS:

Concern to RDBMS there are 5-types of joins:

- **Equi-Join:** Combines common records from two tables based on equality condition. Technically, Join made by using equality-operator (=) to compare values of PrimaryKey of one table and Foriegn Key values of antoher table, hence result set includes common(matched) records from both tables. For implementation see INNER-JOIN.
- **Natural-Join:** It is enhanced version of Equi-Join, in which SELECT operation omits duplicate column. For implementation see INNER-JOIN
- **Non-Equi-Join:** It is reverse of Equi-join where joining condition is uses other than equal operator(=) e.g, !=, <=, >=, >, < or BETWEEN etc. For implementation see INNER-JOIN.
- **Self-Join:**: A customized behavior of join where a table combined with itself; This is typically needed for querying self-referencing tables (or Unary relationship entity). For implementation see INNER-JOINs.
- **Cartesian Product:** It cross combines all records of both tables without any condition. Technically, it returns result set of a query without WHERE-Clause.

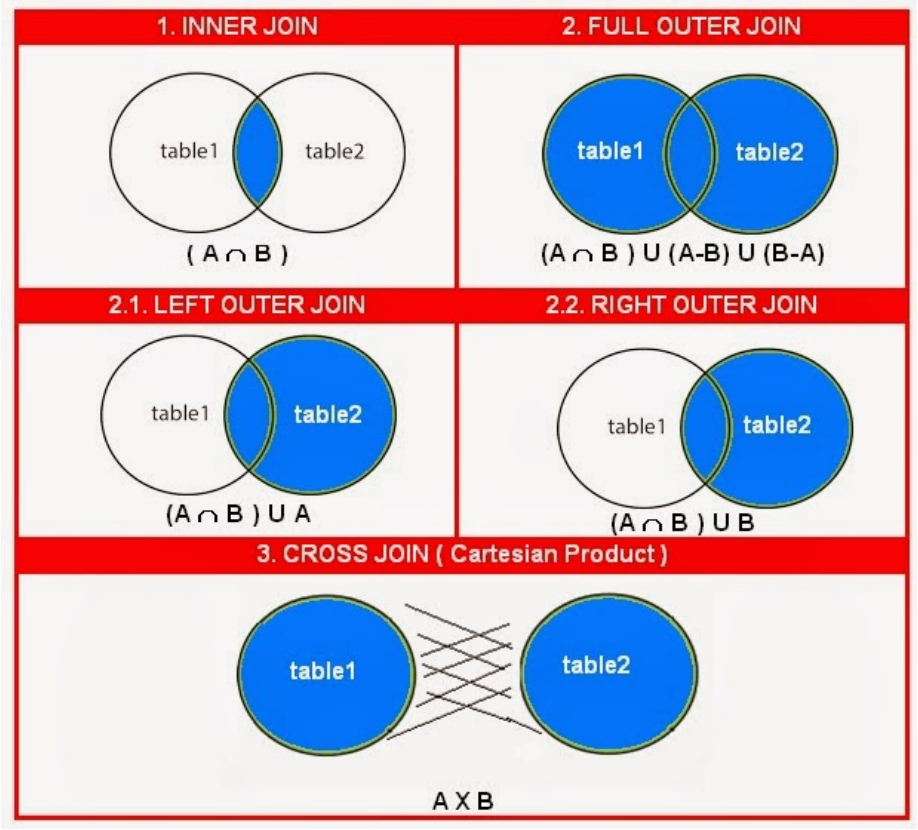
As per SQL concern and advancement, there are 3-types of joins and all RDBMS joins can be achvied using these types of joins.

1. **INNER-JOIN:** It merges(or combiens) matched rows from two tables. The matching is done based on common columns of tables and their comparing operation. If equaility based condition then: EQUI-JOIN performed, otherwise Non-EQUI-Join.
2. **OUTER-JOIN:**It merges(or combines) matched rows from two tables and unmatched rows with NULL values. However, can customized selection of un-matched rows e.g, selecting unmatched row from first table or second table by sub-types: LEFT OUTER JOIN and RIGHT OUTER JOIN.

   2.1. **LEFT Outer JOIN** (a.k.a, LEFT-JOIN): Returns matched rows form two tables and unmached from LEFT table(i.e, first table) only.

   2.2. **RIGHT Outer JOIN** (a.k.a, RIGHT-JOIN): Returns matched rows from two tables and unmatched from RIGHT table only.

   2.3. **FULL OUTER JOIN** (a.k.a OUTER JOIN): Returns matched and unmatched from both tables.
3. **CROSS-JOIN:** This join does not merges/combiens instead it performs cartisian product.



Note: Self-JOIN can be achived by either INNER-JOIN, OUTER-JOIN and CROSS-JOIN based on requirement but table must join with itself.

For more information:

Examples:

1.1: INNER-JOIN: Equi-join implemetation

```
SELECT  *
FROM Table1 A
 INNER JOIN Table2 B ON A.<PrimaryKey> =B.<ForeignKey>;
```

1.2: INNER-JOIN: Natural-JOIN implementation

```
Select A.*, B.Col1, B.Col2          --But no B.ForiengKyeColumn in Select
 FROM Table1 A
 INNER JOIN Table2 B On A.Pk = B.Fk;
```

1.3: INNER-JOIN with NON-Eqijoin implementation

```
Select *
 FROM Table1 A INNER JOIN Table2 B On A.Pk <= B.Fk;
```

1.4: INNER-JOIN with SELF-JOIN

```
Select *
 FROM Table1 A1 INNER JOIN Table1 A2 On A1.Pk = A2.Fk;
```

2.1: OUTER JOIN (full outer join)

```
Select *
 FROM Table1 A FULL OUTER JOIN Table2 B On A.Pk = B.Fk;
```

2.2: LEFT JOIN

```
Select *
 FROM Table1 A LEFT OUTER JOIN Table2 B On A.Pk = B.Fk;
```

2.3: RIGHT JOIN

```
Select *
 FROM Table1 A RIGHT OUTER JOIN Table2 B On A.Pk = B.Fk;
```

3.1: CROSS JOIN

```
Select *
 FROM TableA CROSS JOIN TableB;
```

3.2: CROSS JOIN-Self JOIN

```
Select *
 FROM Table1 A1 CROSS JOIN Table1 A2;
```

//OR//

```
Select *
 FROM Table1 A1,Table1 A2;
```

edited May 14 at 11:07
Community ♦
1

answered Jan 13 at 6:57
nayeemDotNetAuthorities
123 3

3 in your diagrams left jon and right join same.... change it – Ritabrata Gautam Mar 17 at 20:37

can be *achvied ACHIEVED – KNU May 19 at 14:21

good answer it was really hepful... – 404 May 26 at 9:15

add comment

---

protected by Brad Larson ♦ Apr 25 at 19:41

Thank you for your interest in this question. Because it has attracted low-quality answers, posting an answer now requires 10 reputation on this site.

Would you like to answer one of these unanswered questions instead?

Not the answer you're looking for? Browse other questions tagged sql database oracle postgresql rdbms or ask your own question.